Superdevelopments for Weak Reduction*

Eduardo Bonelli
CONICET and Universidad Nacional de Quilmes (Argentina)
ebonelli@unq.edu.ar

Pablo Barenbaum
Universidad de Buenos Aires (Argentina)
foones@gmail.com

We study superdevelopments in the weak lambda calculus of Çağman and Hindley, a confluent variant of the standard weak lambda calculus in which reduction below lambdas is forbidden. In contrast to developments, a superdevelopment from a term M allows not only residuals of redexes in M to be reduced but also some newly created ones. In the lambda calculus there are three ways new redexes may be created; in the weak lambda calculus a new form of redex creation is possible. We present labeled and simultaneous reduction formulations of superdevelopments for the weak lambda calculus and prove them equivalent.

1 Introduction

In contrast to λ-calculus, which allows reduction under the lambda, weak λ-calculus does not. This results in a calculus which is arguably more relevant to programming languages given that the latter consider abstractions as values. However, simply dropping the reduction scheme:

$$\frac{M \to N}{\lambda x.M \to \lambda x.N}\,\xi$$

causes confluence to fail, as may be easily verified. A restriction of the ξ-scheme recovers confluence [4, 11]. Here, the judgement $M \xrightarrow{\Delta} N$ means "M reduces to N by contracting redex Δ" and $\mathsf{fv}(\Delta)$ are the free variables of Δ:

$$\frac{M \xrightarrow{\Delta} N \quad x \notin \mathsf{fv}(\Delta)}{\lambda x.M \xrightarrow{\Delta} \lambda x.N}\,\xi'$$

The resulting weak λ calculus (λ^w-calculus) enjoys *finite developments*: all developments are finite and end in the same term. A development from a term M is a reduction sequence in which only residuals of redexes present in M are reduced. In this paper we study *superdevelopments* [2, 15] in the λ^w-calculus. A superdevelopment allows not only residuals of redexes in M to be reduced but also those upward created ones in the sense that the created redex occurs at a prefix of the reduced redex. There are three ways in which a redex may be created in λ-calculus [10]:

I. $(\lambda x.x)\,(\lambda y.P)\,Q \to (\lambda y.P)\,Q$

II. $(\lambda x.\lambda y.P)\,R\,Q \to (\lambda y.P\{x := R\})\,Q$

III. $(\lambda x.C[x\,Q])\,\lambda y.P \to C'[(\lambda y.P)\,Q']$, where $C' = C\{x := \lambda y.P\}$ and $Q' = Q\{x := \lambda y.P\}$.

A superdevelopment from M allows contraction of newly created redexes of type I and II (i.e. upward creation). In λ^w-calculus we meet two differences. First, redex creation of type III is restricted to those cases in which Q does not have free occurrences of variables bound in C above the hole. Second, there is a new way of creating redexes (hence redex creation in λ^w-calculus is not derived from that of λ-calculus):

*Work partially supported by Instituto Tecnológico de Buenos Aires and LIFIA

M. Fernandez (Ed.): 9th Int. Workshop on Reduction Strategies
in Rewriting and Programming (WRS'09)
EPTCS 15, 2010, pp. 20–31, doi:10.4204/EPTCS.15.2

IV. $(\lambda x.C[(\lambda y.P)\,Q])R \to C'[(\lambda y.P')\,Q']$, where $x \in \mathrm{fv}((\lambda y.P)\,Q)$ and no free variables in $(\lambda y.P)\,Q$ are bound in C; $C' = C\{x := R\}$ and $Q' = Q\{x := R\}$ and $P' = P\{x := R\}$.

In the reduction step $(\lambda x.I\,x)\,y \to \underline{I\,y}$, where $I = \lambda x.x$, the underlined redex is a new redex of type IV.

We define weak superdevelopments (i.e. superdevelopments in λ^w-calculus) by means of an appropriate labeling scheme. Although attractive due to its conciseness, this definition requires reducing terms to *normal form* by means of this notion of labeled reduction. More convenient is a direct inductive definition. Therefore, we introduce a notion of simultaneous reduction, similar to simultaneous (a.k.a. parallel) reduction in λ-calculus. The topic of this paper is to exhibit the complications arising when one tries to prove these notions equivalent and the approach we take to resolve these issues.

Motivation. The starting point of this work is an attempt at extending the concept of orthogonal systems and the confluence results of Mayr and Nipkow [12] to weak higher-order rewriting. Orthogonality depends on whether weak or strong reduction is considered. For eg. in weak reduction $\beta\eta$ as an HRS [12] has only one critical pair (in contrast to the two critical pairs that arise under strong reduction). Moreover, some systems such as $\{g(\lambda x.f(x)) \to a, f(y) \to b\}$ are not orthogonal for strong reduction but are for weak reduction. Another difference between weak and strong reduction lies in standardization [1]. Standardization results for strong reduction in higher-order rewriting, such as HRS, apply to left-linear, pattern HRS that are *fully-extended* (roughly that redexes behave as in the first-order case - they are determined exclusively by their term structure). For instance, the first rule in the HRS $\{f(\lambda x.y) \to g(y), h(x) \to a\}$ is not fully extended since in order to determine if a term of the form $f(\lambda x.M)$ is a redex, it must be checked whether x occurs free in M or not. Fully-extendedness is needed in order for anti-standard pairs to be swappable and hence non-standard reductions to be standardized. For e.g. in $f(\lambda x.h(x)) \to f(\lambda x.a) \to g(a)$ the first and the second steps cannot be swapped. Note, however, that in weak reduction fully-extendedness is not required: indeed, the first rewrite step in the derivation is not a valid weak step. Returning to our initial motivation on confluence for orthogonal weak HRS, the proof of Mayr and Nipkow [12] resorts to Aczel's notion of superdevelopments (but for HRS). As will be explained in this work, a formalization of weak superdevelopments in terms of simultaneous reduction for HRS is involved. The reason is that it requires allowing some (but not all) redexes to be reduced *under* abstractions. For eg. in the HRS $\{f(\lambda x.g(yx)) \to ya, k(x) \to x\}$, the term $f(\lambda x.g(k(x)))$ weakly superdevelops to a (note that the redex $k(x)$ is allowed to be reduced), however $f(\lambda x.k(g(x)))$ does not weakly superdevelop to a (the redex $k(g(x))$ is not allowed to be reduced). Therefore, in order to get a clear grasp of the problem we choose to first address this task for the λ^w-calculus. It should be mentioned that proofs of confluence for weak, orthogonal HRS that rely on developments rather than superdevelopments should go through. However, a number of applications of superdevelopments to confluence, normalisation and higher-order matching, as discussed below, suggest that the concept of superdevelopment deserves to be studied in its own right.

Related work. According to Çağman and Hindley [4] weak reduction, as presented in this work, is due to Howard [8]. It arises as an attempt to construct a tighter correspondence between reduction in Combinatory Logic and β reduction. Çağman and Hindley [4] give a clear account of λ^w-calculus and its relation to β reduction. Lévy and Maranget [11] study a calculus of explicit substitutions for λ^w-calculus, stating that the theory of weak reduction is rather poorly developed in the literature. In a sequel paper [3] they introduce a labeled variant in order to study sharing for this calculus. Notions of weak reduction for calculi with explicit substitutions have been studied by Fernández et al [6, 7]. The latter considers two variants of the *Beta* rule, one in which the argument is required to be closed and one in which the function is required to be closed. This suggests other variations on weak reduction for HRS, although implementation concerns for HRS are out of the scope of this work. Superdevelopments were introduced by Aczel to prove confluence of a generalization of lambda calculus [2]. Van Raamsdonk [15] proves

finiteness of superdevelopments and confluence of orthogonal CRS by adapting Aczel's technique. Two additional different proofs of finiteness are given in [16]. Mayr and Nipkow use a similar technique to prove confluence of orthogonal PRS [12]. Another application of superdevelopments is in higher-order matching. This problem is usually stated in the setting of typed lambda calculus. In order to obtain decidable subclasses of this problem, terms are usually restricted to some particular order. An alternative approach in restricting the problem is to weaken the reduction relation from reduction to normal form to superdevelopments. de Moor and Sittampalam [13, 14, 17, 5] study matching modulo superdevelopments. Khasidashvili and Piperno [9] show that the amount of superdevelopments required to normalize certain classes of terms can be determined statically.

Preliminaries. Assume given a denumerably infinite set of term variables $\mathscr{V}$. The set of (λ^w-calculus) terms Λ and contexts are defined as follows:

$$M ::= x \mid MM \mid \lambda x.M \qquad C ::= \Box \mid CM \mid MC \mid \lambda x.C$$

Free ($\mathsf{fv}(M)$) and bound ($\mathsf{bv}(M)$) variables are as usual. We assume the convention that bound variables are different from free variables and, moreover, bound variables of distinct binders have been renamed apart. Capture avoiding substitution of all free occurrences of x in M by N is written $M\{x := N\}$. In a statement in which distinct variables $x_1,\ldots,x_n$ occur we use $\overline{x}$ for the set $\{x_1,\ldots,x_n\}$. We write $C[M]$ for the result of replacing the hole in C with M (this may bind the free variables of M in $C[M]$). The binding path of C, $\mathsf{bp}(C)$, is the sequence of variables that are bound in C above the hole ($\stackrel{\circ}{=}$ is definitional equality): $\mathsf{bp}(\Box) \stackrel{\circ}{=} \varepsilon$, $\mathsf{bp}(CN) \stackrel{\circ}{=} \mathsf{bp}(C)$, $\mathsf{bp}(MC) \stackrel{\circ}{=} \mathsf{bp}(C)$ and $\mathsf{bp}(\lambda x.C) \stackrel{\circ}{=} \mathsf{bp}(C)\cdot x$. A position (p,q,r) is a sequence of positive integers; ε is the root position (empty sequence) and $p\cdot q$ is the (associative) operation of sequence composition. If P is a set of positions we write $p\cdot P$ for the set resulting from composing p with each position in P. We write $\mathsf{pos}(M)$ for the set of positions of M: $\mathsf{pos}(x) \stackrel{\circ}{=} \{\varepsilon\}$, $\mathsf{pos}(MN) \stackrel{\circ}{=} (0\cdot\mathsf{pos}(M)) \cup (1\cdot\mathsf{pos}(N))$, $\mathsf{pos}(\lambda x.M) \stackrel{\circ}{=} 1\cdot\mathsf{pos}(M)$. The subterm of M at position p is $M|_p$. Also, $M[N]_p$ stands for the term resulting from replacing the subterm at position p in M with N (this may bind the free variables of N in $M[N]_p$). We write $\twoheadrightarrow$ for the reflexive–transitive closure of a binary relation $\rightarrow$. We write S,T for sequences of variables and U,V for sets of variables. Also, $S \oplus T$ is the sequence resulting from concatenating S with T. We say a term M is *away from* a sequence of variables S and write $M \uparrow S$ if $\mathsf{fv}(M)$ is disjoint from S. We write $S \subseteq T$ to indicate that the underlying set of S is included in that of T.

Structure of the paper. Sec. 2 proves that the above mentioned redex creation types are the only possible ones. Sec. 3 introduces two definitions of weak superdevelopments in λ^w-calculus. Sec. 4 addresses the proof of equivalence of these. Finally, we conclude and report on our ongoing work on extensions to higher-order rewriting.

2 Redex Creation in λ^w-calculus

This section characterizes redex creation in λ^w-calculus. In order to follow redexes along reductions we mark them (with a star) and only allow such marked redexes to be contracted. The set of terms $\Lambda^\star$ and contexts of the resulting formalism ($\lambda^w_\star$-calculus) are defined as follows:

$$\begin{array}{rcl} M & ::= & x \mid MM \mid \lambda x.M \mid (\lambda^\star x.M)\,M \\ C & ::= & \Box \mid CM \mid MC \mid \lambda x.C \mid (\lambda^\star x.C)\,M \mid (\lambda^\star x.M)\,C \end{array}$$

The set of positions and binding path is extended accordingly. In the sequel of this subsection, when we speak of "terms" we mean "marked terms" and likewise for contexts. If M is a term and $p \in \mathsf{pos}(M)$,

then we write $\langle M, p\rangle$ for the context resulting from replacing the term at p in M with a hole. If the hole in C is at position p we write $C[]_p$.

As mentioned, reduction in $\lambda^w_\star$-calculus is similar to reduction in λ^w-calculus except only marked redexes may be contracted.

$$\frac{\Delta = (\lambda^\star x.M)\,N}{(\lambda^\star x.M)\,N \xrightarrow{\Delta} M\{x := N\}} \qquad \frac{M \xrightarrow{\Delta} M' \quad \Delta \uparrow \mathsf{bp}(C)}{C[M] \xrightarrow{\Delta} C[M']}$$

We are interested in studying situations where reduction in $\lambda^w_\star$-calculus, from terms where all redexes have been marked, produces terms having occurrences of $(\lambda x.M)\,N$ in contexts with binding paths that they are away from. This models the situation where a new reducible expression, a λ^w-calculus redex, has been produced that was not initially marked.

Definition 2.0.1 *Let $M,N,P \in \Lambda^\star$, $p \in pos(M)$ and $M|_p \uparrow bp(\langle M,p\rangle)$.*

- *If $M|_p = (\lambda^\star x.P)\,N$, then we say $M|_p$ is a $\lambda^w_\star$-calculus redex at (M,p).*
- *If $M|_p = (\lambda x.P)N$, then we say $M|_p$ is a λ^w-calculus redex at (M,p).*

A term $M \in \Lambda^\star$ is *initially marked* iff the set of marked subterms are indeed $\lambda^w_\star$-calculus redexes and all λ^w-calculus redexes have been marked. More precisely, iff the following conditions hold:

1. $\forall p \in \mathsf{pos}(M).M|_p = (\lambda^\star x.P)\,Q$ implies $M|_p$ is a $\lambda^w_\star$-calculus redex at (M,p).
2. $\forall p \in \mathsf{pos}(M).M|_p = (\lambda x.P)\,Q$ implies $M|_p$ is not a λ^w-calculus redex at (M,p).

The following result is proved by case analysis on the relative positions of p and q.

Proposition 2.0.2 (Redex creation) *Let $M \in \Lambda^\star$ be initially marked, $M \xrightarrow{\Delta} N$ and $q \in pos(N)$ s.t. there is a λ^w-calculus redex at (N,q). One of the following situations must arise:*

- ***Case I**: $M = C[(\lambda^\star x.x)\,(\lambda y.M_1)M_2]$ and $N = C[(\lambda y.M_1)\,M_2]_q$.*
- ***Case II**: $M = C[(\lambda^\star x.(\lambda y.M_1))\,Q\,M_2]$ and $N = C[(\lambda y.M_1')M_2]_q$, where $M_1' = M_1\{x := Q\}$.*
- ***Case III**: $M = C_1[(\lambda^\star x.C_2[xM_2])\,\lambda y.M_1]$ and $N = C_1[C_2'[(\lambda y.M_1)M_2']_{q_2}]_{q_1}$, where $q = q_1 \cdot q_2$, $C_2' = C_2\{x := \lambda y.M_1\}$ and $M_2' = B\{x := \lambda y.M_1\}$.*
- ***Case IV**: $M = C_1[(\lambda^\star x.C_2[(\lambda y.M_1)M_2])\,Q]$ and $N = C_1[C_2'[(\lambda y.M_1')M_2']_{q_2}]_{q_1}$, where $q = q_1 \cdot q_2$, $M_1' = M_1\{x := Q\}$, $M_2' = M_2\{x := Q\}$, $C_2' = C_2\{x := Q\}$ and $x \in fv((\lambda y.M_1')M_2')$.*

3 Superdevelopments in λ^w-calculus

This section introduces two presentations of superdevelopments in λ^w-calculus: via labeled reduction (Sec. 3.1) and simultaneous reduction (Sec. 3.2).

3.1 Weak Superdevelopments via Labeled Reduction

We begin by introducing the labeled λ^w-calculus (λ^w_ℓ-calculus). Assume given a denumerably infinite set of *labels* $\mathscr{L}$. The labeled terms Λ_ℓ and contexts are given by the following grammar:

$$A ::= x \mid \lambda^a x.A \mid @(A^a, A) \qquad C ::= \Box \mid \lambda^a x.C \mid @(C^a, A) \mid @(A^a, C)$$

where $x \in \mathscr{V}$ and $a \in \mathscr{L}$. Ocassionally, we write $\lambda^{a_1 \ldots a_n} x_1 \ldots x_n.A$ (or simply $\lambda^{\overline{a}^n} \overline{x}^n.A$) as a shorthand for $\lambda^{a_1} x_1.\lambda^{a_2} x_2. \ldots \lambda^{a_n} x_n.A$. If we wish to single out the leftmost abstraction we write $\lambda^{a\overline{a}^n} x\overline{x}^n.A$. Thus abstractions and (the first argument of) applications are decorated with labels. In $@(A^a,B)$ the depicted label binds all the occurrences of a in A. The set of free labels of a labeled term is defined as follows:

$$\begin{array}{rcl} \mathsf{fl}(x) & \doteq & \emptyset \\ \mathsf{fl}(\lambda^a x.A) & \doteq & \{a\} \cup \mathsf{fl}(A) \\ \mathsf{fl}(@(A^a,B)) & \doteq & \mathsf{fl}(A) \setminus \{a\} \cup \mathsf{fl}(B) \end{array}$$

We assume the existence of a distinguished label $\star \in \mathscr{L}$ which is never bound. Also, $M_{\{-a\}}$ denotes the substitution of all free occurrences of label a with $\star$, and $|A| \in \Lambda$ is the term resulting from A by erasing all labels (and identifying $@(A,B)$ with AB), in which case we say A is a *labeling* of $|A|$. Substitution over labeled terms is written $A\{x := B\}$. Note that this operation must not capture labels. For example, $@(x^a,x)\{x := \lambda^a y.y\} \neq @((\lambda^a y.y)^a,(\lambda^a y.y))$, rather the binding occurrence of a in $@(x^a,x)$ must first be renamed to $@(x^b,x)$ in order for substitution to yield $@(x^b,x)\{x := \lambda^a y.y\} = @((\lambda^a y.y)^b,(\lambda^a y.y))$. The binding path of a labeled context C is the binding path of $|C|$.

Remark 3.1.1 *Labeled characterizations of superdevelopments we know of do not bind labels in applications. Instead they define a term to be good [15]*[1] *or well-labeled [16, 5] if an occurrence of a label a decorates an application and another occurrence of a decorates an abstraction, then this abstraction must occur in one of the arguments of the application. It is stated that reduction preserves well-labeledness, however this in fact fails. Eg. taking $A \doteq @((\lambda^a y.y)^a,z)$, clearly the reduction step $@((\lambda^c x.@(x^b,x))^c,A) \to @(A^b,A)$ produces a non well-labeled term. Note that the results in op.cit. still hold (except for preservation of well-labeledness under reduction, as illustrated) since, by well-labeledness, these copies of A never interact with one another.*

A term A is said to be *initially labeled* iff all abstractions have distinct labels. Note that, given a term A, it is always possible to produce a label a that does not occur in it given that A is finite and $\mathscr{L}$ is not. Reduction in λ^w_ℓ-calculus is defined below, where S is a sequence of variables and the depicted occurrence of $@((\lambda^a x.A)^a,B)$ is called a *redex*.

$$\frac{@((\lambda^a x.A)^a,B) \uparrow \mathsf{bp}(C) \oplus S}{C[@((\lambda^a x.A)^a,B)] \xrightarrow{S}_\ell C[A_{\{-a\}}\{x := B\}]}$$

We substitute a with $\star$ to avoid bound labels from becoming free as in $@((\lambda^a x.\lambda^a y.y)^a,w) \xrightarrow{S}_\ell \lambda^a y.y$ and, consequently to avoid rebinding of labels such as in $@(\underline{@((\lambda^a x.\lambda^a y.y)^a,w)}^a,z) \xrightarrow{S}_\ell @(\lambda^a y.y^a,z)$. We write $\to_\ell$ for $\xrightarrow{\emptyset}_\ell$. Note that $A \xrightarrow{S}_\ell B$ and $T \subseteq S$ implies $A \xrightarrow{T}_\ell B$. Also, reduction does not create free variables. In the judgement $A \xrightarrow{S}_\ell B$ we implicitly assume that $A,B \in \Lambda_\ell$.

Definition 3.1.2 *M weakly superdevelops to N if there exists an initially labeled term A and a labeled term B s.t. $|A| = M$ and $|B| = N$ and $A \twoheadrightarrow_\ell B$. If, moreover, B is in $\to_\ell$-normal form, then we say there is a* complete *weak superdevelopment from M to N.*

Some properties of labeled reduction are considered below. Since weak superdevelopments are also superdevelopments (which are finite [15, 16]):

[1] This work decorates applications with *sets* of labels, however the example of this remark still applies by considering $\{a\}$ instead of a where appropriate.

Lemma 3.1.3 *Weak superdevelopments are finite.*

A further property is that S may be weakened with further variables not occurring free in A without affecting reduction. This extends to many-step reduction.

Lemma 3.1.4 *$A \xrightarrow{S}_\ell B$ and $x \notin fv(A)$ implies $A \xrightarrow{S \oplus x}_\ell B$.*

Regarding reduction and the context in which the redex occurs:

Lemma 3.1.5 *The following are equivalent:*

1. *Suppose $A \to_\ell B$ by contracting a redex Δ. $\Delta \uparrow S$ iff $C[A] \to_\ell C[B]$ for every context C s.t. $bp(C) = S$.*
2. *$C_1[C_2[\Delta]] \xrightarrow{S}_\ell C_1[C_2[\Delta']]$ iff $C_2[\Delta] \xrightarrow{S \oplus bp(C_1)}_\ell C_2[\Delta']$, where Δ is the contracted redex.*

Finally, we prove that substitution preserves reduction (Lem. 3.1.6(2)). The proof is by induction on A, resorting to Lem. 3.1.5(2) and Lem. 3.1.6(1).

Lemma 3.1.6
1. *If $a \notin fl(B)$, $A_{\{-a\}}\{x := B\} = A\{x := B\}_{\{-a\}}$.*
2. *Suppose $A \xrightarrow{S}_\ell A'$ and $B \uparrow S$. Then $A\{x := B\} \xrightarrow{S}_\ell A'\{x := B\}$.*

Proof. Both proofs procceed by induction on A. In the cases where a variable y is bound in A, we assume the convention that y does not occur in B. Also, when a head reduction takes place, we resort to the fact that $A_1\{x := B\}\{y := A_2\{x := B\}\} = A_1\{y := A_2\}\{x := B\}$.

■

3.2 Weak Superdevelopments via Simultaneous Reduction

An alternative presentation of weak superdevelopments is by means of simultaneous reduction. It has numerous benefits over labeled reduction. One is that it satisfies the diamond property (and can be used for proving confluence of labeled reduction and the λ^w-calculus (Prop. 4.3.6)). Another is that we can avoid reasoning over reduction to a normal form: one simultaneous reduction step suffices for superdeveloping a term. A naive attempt at formalizing this notion in big-step style fails. Let us write $M \overset{S}{\Rightarrow} N$ for such a judgement, where S is a sequence of variables which denotes the binding context (i.e. the variables that are bound in the context) in which this superdevelopment takes place. This judgement would include the inference scheme:

$$\frac{M \overset{S}{\Rightarrow} \lambda x.M' \quad N \overset{S}{\Rightarrow} N' \quad (\lambda x.M')N' \uparrow S}{MN \overset{S}{\Rightarrow} M'\{x := N'\}} \tag{1}$$

However, it turns out that we need to consider an exception to the condition $(\lambda x.M')N' \uparrow S$, namely when an abstraction contributes to a head redex in a superdevelopment. Indeed, in this case the abstracted variable is to be substituted and hence redexes which contain occurrences of this variable *can* be contracted. As an example, consider the following superdevelopment in the λ^w-calculus:

$$I(\lambda x.Ix)y \to (\lambda x.Ix)y \to Iy \to y$$

To deduce the judgement $I(\lambda x.Ix)y \overset{S}{\Rightarrow} y$ using (1) we require $I(\lambda x.Ix) \overset{S}{\Rightarrow} \lambda x.x$. For this to hold we must allow contraction of Ix. As a consequence, we study an extended judgement $M \overset{S,k}{\Rightarrow} N$ in which the integer $k \geq 0$ indicates how many abstractions of M contribute to a head redex in a later stage of the derivation. Reduction under these abstractions is allowed. Note that, in contrast to Sec. 3.1 where S in $A \overset{S}{\to}_\ell B$ includes all the variables in A bound above the redex that is reduced, in the judgement $M \overset{S,k}{\Rightarrow} N$ the sequence S contains only the variables in M bound above the redexes of the superdevelopment under which reduction is not allowed.

Definition 3.2.1 *There is a* superstep from M to N under S,k *iff* $M \overset{S,k}{\Rightarrow} N$*, where this judgement is defined as follows:*

$$\frac{}{x \overset{S,0}{\Rightarrow} x}\ \mathsf{Var} \qquad \frac{M \overset{x\cdot S,0}{\Rightarrow} M'}{\lambda x.M \overset{S,0}{\Rightarrow} \lambda x.M'}\ \mathsf{Abs1} \qquad \frac{M \overset{S,k}{\Rightarrow} M'}{\lambda x.M \overset{S,k+1}{\Rightarrow} \lambda x.M'}\ \mathsf{Abs2} \qquad \frac{M \overset{S,0}{\Rightarrow} M' \quad N \overset{S,0}{\Rightarrow} N'}{MN \overset{S,0}{\Rightarrow} M'N'}\ \mathsf{App1}$$

$$\frac{M \overset{S,n+1}{\Rightarrow} \lambda x.M' \quad N \overset{S,m}{\Rightarrow} N' \quad (\lambda x.M')N' \uparrow S \quad m > 0 \Rightarrow M' = \lambda \overline{x}^n.x}{MN \overset{S,n+m}{\Rightarrow} M'\{x := N'\}}\ \mathsf{App2}$$

There is a complete *superstep from M to N under S,k if in the derivation of the judgement $M \overset{S,k}{\Rightarrow} N$ the scheme* App1 *is used only if* App2 *is not applicable.*

Note that this definition establishes an inside-out strategy for computing a complete weak superdevelopments.

4 Equivalence of Presentations

In this section we prove the following result (the first item in Sec. 4.2 and the second in Sec. 4.3), where we write M_ℓ for any labeling of M.

Theorem 4.0.2 *1. If $M \overset{S,0}{\Rightarrow} N$, then there exist M_ℓ, N_ℓ s.t. $M_\ell \overset{S}{\twoheadrightarrow}_\ell N_\ell$.*

2. If $M_\ell \overset{S}{\twoheadrightarrow}_\ell N_\ell$ and N_ℓ is in normal form, then $M \overset{S,0}{\Rightarrow} N$.

In the second item, note that the binding nature of labels in applications is required for the statement to hold. For example, this is the reduction sequence one would obtain were labels in applications not considered binding:

$$@(\lambda^b x.@(x^a,y)^b, \lambda^a z.z) \to_\ell @((\lambda^a z.z)^a, y) \to_\ell y$$

Notice that it is not the case that $@(\lambda^b x.@(x^a,y)^b, \lambda^a z.z) \overset{S,0}{\Rightarrow} y$ as may be seen by trying to derive this judgement. The requirement that N_ℓ be in normal form is justified by the following example, where A is any redex s.t. $A \to_\ell A'$:

$$@((\lambda^a x.@(x^b,x))^a, A) \to_\ell @(A^b, A) \to_\ell @(A'^b, A).$$

It is clear that labeled reduction still has some work to do: a labeled redex remains (i.e. it is an incomplete weak superdevelopment). In fact, this is an example of an incomplete development. For this reason, the judgement $@((\lambda^a x.@(x^b,x))^a, A) \overset{S,0}{\Rightarrow} AA'$ is not derivable.

4.1 Supersteps over Labeled Terms

As mentioned, we have to relate labeled reduction to *normal form* with supersteps. In order to do so, we introduce an intermediate notion of supersteps over *labeled* terms $A \overset{S,k}{\Rightarrow}_\ell B$. The reason is that when passing from $\to_\ell$ steps to $\overset{S,k}{\Rightarrow}$ steps we lose the labels and hence our handle over this normal form (which is the complete weak superdevelopment of the *labeled* redexes). In summary, in Sec. 4.2, in relating $\overset{S,k}{\Rightarrow}$ with $\twoheadrightarrow_\ell$, we shall first go from $\overset{S,k}{\Rightarrow}$ to $\overset{S,k}{\Rightarrow}_\ell$ (defined below) and then to $\twoheadrightarrow_\ell$. Conversely, in Sec. 4.3, we shall first go from $\twoheadrightarrow_\ell$ to $\overset{S,k}{\Rightarrow}_\ell$ and then to $\overset{S,k}{\Rightarrow}$.

Definition 4.1.1 (Supersteps over labeled terms) *We say there is a* superstep from A to B under S,k *iff* $A \overset{S,k}{\Rightarrow}_\ell B$*, where this judgement is defined as follows:*

$$\frac{}{x \overset{S,0}{\Rightarrow}_\ell x}\ \mathsf{LVar} \qquad \frac{A \overset{x\cdot S,0}{\Rightarrow}_\ell A'}{\lambda^a x.A \overset{S,0}{\Rightarrow}_\ell \lambda^a x.A'}\ \mathsf{LAbs1}$$

$$\frac{A \overset{S,k}{\Rightarrow}_\ell A'}{\lambda^a x.A \overset{S,k+1}{\Rightarrow}_\ell \lambda^a x.A'}\ \mathsf{LAbs2} \qquad \frac{A \overset{S,0}{\Rightarrow}_\ell A' \quad B \overset{S,0}{\Rightarrow}_\ell B'}{@(A^a,B) \overset{S,0}{\Rightarrow}_\ell @(A'^a,B')}\ \mathsf{LApp1}$$

$$\frac{A \overset{S,n+1}{\Rightarrow}_\ell \lambda^a x.A' \quad B \overset{S,m}{\Rightarrow}_\ell B' \quad @((\lambda^a x.A')^a,B') \uparrow S \quad m>0 \Rightarrow A' = \lambda^{\overline{a}^n}\overline{x}^n.x}{@(A^a,B) \overset{S,n+m}{\Rightarrow}_\ell A'_{\{-a\}}\{x := B'\}}\ \mathsf{LApp2}$$

Note that $\overset{S,k}{\Rightarrow}_\ell$ is reflexive and $A \overset{S,k}{\Rightarrow}_\ell B$ implies $\mathsf{fv}(A) \supseteq \mathsf{fv}(B)$. Additional basic properties of reduction are stated below. The first one states that a weak superdevelopment under S,k leaves k abstractions at the root of the term. The remaining ones indicate how labeled and unlabeled simultaneous reduction relate.

Lemma 4.1.2 *1. If $A \overset{S,k}{\Rightarrow}_\ell B$, then there exist variables x_i and labels a_i, with $1 \le i \le k$, and $B' \in \Lambda_\ell$ s.t. $B = \lambda^{a_1 \dots a_k} x_1 \dots x_k.B'$.*

2. If $A \overset{S,k}{\Rightarrow}_\ell B$, then $|A| \overset{S,k}{\Rightarrow} |B|$.

3. If $M \overset{S,k}{\Rightarrow} N$, then there exist $M_\ell, N_\ell \in \Lambda_\ell$ s.t. $M_\ell \overset{S,k}{\Rightarrow}_\ell N_\ell$.

4.2 From Supersteps to Labeled Reduction

We address the proof of the first item of Thm. 4.0.2: If $M \overset{S,0}{\Rightarrow} N$, then there exist M_ℓ, N_ℓ s.t. $M_\ell \overset{S}{\twoheadrightarrow}_\ell N_\ell$. Note that from $M \overset{S,k}{\Rightarrow} N$ and Lem. 4.1.2(3), there exist M_ℓ, N_ℓ s.t. $M_\ell \overset{S,k}{\Rightarrow}_\ell N_\ell$. Thus we are left to verify that

$$M_\ell \overset{S,k}{\Rightarrow}_\ell N_\ell \text{ implies } M_\ell \overset{S}{\twoheadrightarrow}_\ell N_\ell \tag{2}$$

In general, (2) does not hold. The problem is that $k > 0$ allows redexes with occurrences of bound variables to be contracted. Eg. $\lambda^a x.@((\lambda^b y.y)^b,x) \overset{\varepsilon,1}{\Rightarrow}_\ell \lambda^a x.x$ but $\lambda^a x.@((\lambda^b y.y)^b,x) \overset{\varepsilon}{\not\twoheadrightarrow}_\ell \lambda^a x.x$. An attempt to prove (2) by induction on the derivation of the judgement $M_\ell \overset{S,k}{\Rightarrow}_\ell N_\ell$ reveals that we need to

consider a relaxed notion of labeled reduction in which contraction of *some* redexes having free occurrences of bound variables is admitted. The *only* such redexes which are allowed to be contracted are those that contribute to the patterns of redexes that are to be consumed in a weak superdevelopment. We call this *chain reduction*. Its definition arises from a fine analysis of the generalization required of the hypothesis in order for the inductive proof of (2) to go through (particularly when the term is an application).

Definition 4.2.1 (Chain reduction) *The judgement $A \overset{S,k}{\rightsquigarrow} B$ is defined by induction on k.*

- *$A \overset{S,0}{\rightsquigarrow} B$ holds iff $A \overset{S}{\rightarrow\!\!\!\!\rightarrow}_\ell B$.*
- *$A \overset{S,k+1}{\rightsquigarrow} B$ holds iff there exist A_1, A_2 s.t. (1) $A \overset{S}{\rightarrow\!\!\!\!\rightarrow}_\ell \lambda^a x.A_1$; (2) $A_1 \overset{S,k}{\rightsquigarrow} A_2$; and (3) $B = \lambda^a x.A_2$.*

Note that if $A \overset{S,k}{\rightsquigarrow} B$, then there exist variables x_i and labels a_i, with $1 \leq i \leq k$, and $B' \in \Lambda_\ell$ s.t. $B = \lambda^{\overline{a}^k}\overline{x}^k.B'$.

The following congruence properties of chain reduction shall be required. The proof of those for application resort to Lem. 3.1.4, Lem. 3.1.6(2) and the fact that $A \overset{S}{\rightarrow\!\!\!\!\rightarrow}_\ell B$ implies $\mathsf{fv}(A) \cap S = \mathsf{fv}(B) \cap S$.

Lemma 4.2.2 (Abstraction) *1. If $B \overset{x\cdot S,0}{\rightsquigarrow} B'$, then $\lambda^a x.B \overset{S,0}{\rightsquigarrow} \lambda^a x.B'$.*

2. If $B \overset{S,k}{\rightsquigarrow} B'$, then $\lambda^a x.B \overset{S,k+1}{\rightsquigarrow} \lambda^a x.B'$.

Lemma 4.2.3 (Application I) *If $A \overset{S,0}{\rightsquigarrow} A'$ and $B \overset{S,0}{\rightsquigarrow} B'$, then $@(A^a, B) \overset{S,0}{\rightsquigarrow} @(A'^a, B')$.*

Lemma 4.2.4 (Application II) *1. Let $A \overset{S,n+1}{\rightsquigarrow} \lambda^a x.A' = \lambda^{a\overline{a}^n} x\overline{x}^n.x$ and $B \overset{S,m}{\rightsquigarrow} \lambda^{\overline{b}^m}\overline{y}^m.B'$. Assume, moreover, $\lambda^a x.A' \uparrow S$ and $B' \uparrow S$. Then $@(A^a, B) \overset{S,n+m}{\rightsquigarrow} \lambda^{\overline{a}^n\overline{b}^m}\overline{x}^n\overline{y}^m.B'$.*

2. Let $A \overset{S,k+1}{\rightsquigarrow} \lambda^a x.A' = \lambda^{a\overline{a}^k} x\overline{x}^k.A''$ and $B \overset{S,0}{\rightsquigarrow} B'$. Assume, moreover, $\lambda^a x.A' \uparrow S$ and $B' \uparrow S$. Then $@(A^a, B) \overset{S,k}{\rightsquigarrow} \lambda^{\overline{a}^k}\overline{x}^k.A''\{x := B'\}$.

We can now replace (2) by the following statement.

Proposition 4.2.5 *$A \overset{S,k}{\Rightarrow}_\ell B$ implies $A \overset{S,k}{\rightsquigarrow} B$.*

Proof. By induction on the derivation of $A \overset{S,k}{\Rightarrow}_\ell B$. Each case is straightforward, resorting to Lem. 4.2.2 for the rules LAbs1 and LAbs2, Lem. 4.2.3 for the rule LApp1 and Lem. 4.2.4 for the rule LApp2. ∎

The proof of Thm. 4.0.2(1) proceeds as follows. From $M \overset{S,0}{\Rightarrow} N$ and Lem. 4.1.2(3), there exist M_ℓ, N_ℓ s.t. $M_\ell \overset{S,0}{\Rightarrow}_\ell N_\ell$. From Prop. 4.2.5(1), $M_\ell \overset{S,0}{\rightsquigarrow} N_\ell$. Finally, from the definition of chain reduction, $M_\ell \overset{S}{\rightarrow\!\!\!\!\rightarrow}_\ell N_\ell$. The second item of Prop. 4.2.5 will be used in the next section.

Corollary 4.2.6 $\overset{S}{\rightarrow}_\ell \subseteq \overset{S,0}{\Rightarrow}_\ell \subseteq \overset{S}{\rightarrow\!\!\!\!\rightarrow}_\ell$.

Proof.

1. Suppose $A \overset{S}{\rightarrow}_\ell B$. By induction on A follows $A \overset{S,0}{\Rightarrow}_\ell B$.
2. Suppose $A \overset{S,0}{\Rightarrow}_\ell B$. From Prop. 4.2.5, $A \overset{S,0}{\rightsquigarrow} B$. By Def. 4.2.1, this implies $A \overset{S}{\rightarrow\!\!\!\!\rightarrow}_\ell B$.

∎

4.3 From Labeled Reduction to Supersteps

A proof of Thm. 4.0.2(2) requires reasoning over the more general judgement $M_\ell \overset{S}{\twoheadrightarrow}_\ell B$ in which B is not necessarily in normal form but which is related to N_ℓ in the sense that $B \overset{S}{\twoheadrightarrow}_\ell N_\ell$. In turn, for this it is convenient to have a direct inductive characterization of the form of N_ℓ in terms of M_ℓ (cf. Prop 4.3.6). The following notion of full weak superdevelopment provides such a definition.

Definition 4.3.1 (Full weak superdevelopment) *The full weak superdevelopment of A under S,k, written $A^*_{S,}$, is defined by induction on A as follows:*

$$\begin{array}{rcl} x^*_{S,0} & \triangleq & x \\ (\lambda^a x.A)^*_{S,0} & \triangleq & \lambda^a x.A^*_{x\cdot S,0} \\ (\lambda^a x.A)^*_{S,k+1} & \triangleq & \lambda^a x.A^*_{S,k} \\ @(A^a,B)^*_{S,k} & \triangleq & \begin{cases} A'_{\{-a\}}\{x := B'\}, & \text{if } (\star) \\ @({A^*_{S,0}}^a, B^*_{S,0}), & \text{if not } (\star) \text{ and } k=0 \end{cases} \end{array}$$

where $(\star)$ is the following condition: for some $n,m \geq 0$:

1. $A^*_{S,n+1} = \lambda^a x.A' = \lambda^{aa_1\ldots a_n} x x_1 \ldots x_n.A''$
2. $B^*_{S,m} = B' = \lambda^{b_1\ldots b_m} y_1 \ldots y_m.B''$
3. $n+m=k$
4. $m>0 \Rightarrow A'' = x$
5. $@((\lambda^a x.A')^a, B') \uparrow S$.

Remark 4.3.2 *$A^*_{S,k}$ may be undefined for $k>0$. For example, $@(x^a,y)^*_{S,1}$ is undefined for any S given that the full weak superdevelopment of $@(x^a,y)$ does not produce an abstraction. Note, however, that $A^*_{S,0}$ always exists. In general, if $A^*_{S,k}$ is defined, then there exist variables x_i and labels a_i, with $1 \leq i \leq k$, and $A' \in \Lambda_\ell$ s.t. $A^*_{S,k} = \lambda^{a_1\ldots a_k} x_1 \ldots x_k.A'$. Compare this with Lem. 4.1.2(1).*

Lemma 4.3.3 *If there exists an A' such that $A^*_{S,k} = A'$, it is unique.*

Some basic properties of this notion follow.

Lemma 4.3.4 *If $A^*_{S,k}$ exists:*

1. $A \overset{S,k}{\Rightarrow}_\ell A^*_{S,k}$ *when $A^*_{S,k}$ exists.*
2. $A \overset{S}{\twoheadrightarrow}_\ell A^*_{S,0}$.

Proof. The first item is proved by induction on A. The second follows from the first item and Cor. 4.2.6. ∎

We now relate full weak superdevelopments with simultaneous labeled weak superdevelopments.

Lemma 4.3.5 1. *Let $A,B \in \Lambda_\ell$, $S \subseteq \mathscr{V}$, $x \in \mathscr{V}$ and $k \geq 0$ s.t. $x \notin S \cap fv(A)$, $x \notin fv(B)$ and $B \uparrow S$. Then $(A\{x := B\})^*_{S,k}$ exists iff the following conditions hold:*
   (a) There exist $n,m \geq 0$ s.t. $k = n+m$,

(b) $A^*_{S,n}$ *and* $B^*_{S,m}$ *exist,*

(c) $m > 0$ *implies* $A^*_{S,n}$ *has the form* $\lambda^{a_1\dots a_n}x_1\dots x_n.x$.

Moreover, in that case, $(A\{x := B\})^*_{S,k} = A^*_{S,n}\{x := B^*_{S,m}\}$.

2. *If* $A,B \in \Lambda_\ell$, $S_1,S_2 \subseteq \mathscr{V}$, $k_1,k_2 \geq 0$ *s.t.* $S_1 \supseteq S_2$, $k_1 \leq k_2$ *and* $A \overset{S_1,k_1}{\Rightarrow}_\ell B$, *then* $A^*_{S_2,k_2}$ *exists iff* $B^*_{S_2,k_2}$ *exists, and, if they exist, they coincide.*

3. *Let* $A_i \in \Lambda_\ell$ *for* $1 \leq i \leq n$, $S,K \subseteq \mathscr{V}$ *and* $A_i \overset{S,k}{\Rightarrow}_\ell A_{i+1}$ *for all* $1 \leq i < n$. *Then* ${A_1}^*_{S,k}$ *exists iff* ${A_n}^*_{S,k}$ *exists, and, if they exist, they coincide.*

Proof. The first item is proved by induction on A and resorting to the fact that $A^*_{S,k} = A^*_{x\cdot S,k}$ if $x \notin \mathrm{fv}(A)$. The second item is by induction on A and resorts to the first one. The last one is a consequence of the second. ∎

Proposition 4.3.6 λ^w*-calculus is confluent.*

Proof. Lem. 4.3.4(1) and Lem. 4.3.5(2) entail the diamond property of $\overset{S,0}{\Rightarrow}_\ell$. This in turn entails confluence of λ^w-calculus by Cor. 4.2.6. ∎

Lemma 4.3.7 $A \overset{S}{\twoheadrightarrow}_\ell B$ *implies* $B \overset{S}{\twoheadrightarrow}_\ell A^*_{S,0}$.

Proof. Suppose $A \overset{S}{\twoheadrightarrow}_\ell B$. By Cor. 4.2.6, $A \overset{S,0}{\Longrightarrow}_\ell B$. From Lem. 4.3.5 we deduce $A^*_{S,0} = B^*_{S,0}$. Moreover, $B \overset{S}{\twoheadrightarrow}_\ell B^*_{S,0} = A^*_{S,0}$ from Lem. 4.3.4(2). ∎

Regarding the second item of Thm. 4.0.2, suppose $M_\ell \overset{S}{\twoheadrightarrow}_\ell N_\ell$ with N_ℓ in normal form. From Lem. 4.3.7, $N_\ell \overset{S}{\twoheadrightarrow}_\ell {M_\ell}^*_{S,0}$. Moreover, since N_ℓ is in normal form, it coincides with the complete weak superdevelopment of M_ℓ, namely $N_\ell = {M_\ell}^*_{S,0}$. We conclude by resorting to Lem. 4.3.4(1) and then Lem. 4.1.2(2) to deduce $M \overset{S,0}{\Rightarrow} |{M_\ell}^*_{S,0}| = N$.

5 Conclusions

Redex creation in λ^w-calculus is more subtle than in λ-calculus. This raises the question on how superdevelopments behave. We present two possible definitions and prove that they are equivalent. The labeled presentation is easy to grasp but complicated to use in proving results (e.g. properties of reduction). Simultaneous reduction is easier for this purpose. However, such an inductive definition requires dealing with reduction under binders of redexes having free occurrences of bound variables which labeled reduction forbids. This makes the correspondence between these notions of reduction (labeled and simultaneous) more demanding to prove, a task we have taken up in this work.

We are currently developing these results in the framework of higher-order rewriting (HOR). A number of issues arise in this extended setting. First we must consider a notion of orthogonal HOR systems for weak reduction, as discussed in the introduction. Second, we have to determine what it means for a variable to be substituted in order for reduction under binders of redexes involving these variables to be allowed. Eg. in $\{f(\lambda x.g(y(x),z(x))) \to f(\lambda x.g(y(a),z(x)))\}$, redexes involving x which occur below y (once this rule is instantiated) should be permitted but not those below z. Last, there is an additional complication that is not apparent in the setting of the lambda calculus. In the judgement $M \overset{S,n+1}{\Rightarrow} \lambda x.M'$

of App2 it turns out that reduction steps involving free occurences of x do not contribute towards the creation of the outermost β redex of App2. This requirement must be made explicit in HOR. The naive generalization of $M \overset{S,n+1}{\Rightarrow} \lambda x.M'$ to HOR should require that the HOR-reduction steps involving free occurrences of x not contribute towards the newly created outermost redex.

Acknowledgements. To the referees for comments that helped improve this paper.

References

[1] (2003): *Term Rewriting Systems*. Cambridge University Press.

[2] Peter Aczel (1978): *A general Church-Rosser theorem*. Technical report, University of Manchester.

[3] Tomasz Blanc, Jean-Jacques Lévy & Luc Maranget (2005): *Sharing in the Weak Lambda-Calculus*. In: Aart Middeldorp, Vincent van Oostrom, Femke van Raamsdonk & Roel C. de Vrijer, editors: *Processes, Terms and Cycles, Lecture Notes in Computer Science* 3838. Springer, pp. 70–87.

[4] Naim Çağman & J. Roger Hindley (1998): *Combinatory Weak Reduction in Lambda Calculus*. *Theor. Comput. Sci.* 198(1-2), pp. 239–247.

[5] Germain Faure (2006): *Matching Modulo Superdevelopments Application to Second-Order Matching*. In: Miki Hermann & Andrei Voronkov, editors: *LPAR, Lecture Notes in Computer Science* 4246. Springer, pp. 60–74.

[6] Maribel Fernández, Ian Mackie & François-Régis Sinot (2005): *Closed reduction: explicit substitutions without alpha-conversion*. *Mathematical Structures in Computer Science* 15(2), pp. 343–381.

[7] Maribel Fernández, Ian Mackie & François-Régis Sinot (2005): *Lambda-Calculus with Director Strings*. *Appl. Algebra Eng. Commun. Comput.* 15(6), pp. 393–437.

[8] William Howard (1970): *Assignment of ordinals to terms for primtive recursive functionals of finite type*. In: A. Kino, J. Myhill & R.E. Vesley, editors: *Intuitionism and proof theory*. North-Holland, pp. 442–478. Proc. Of Conference in Buffalo, USA, 1968.

[9] Zurab Khasidashvili & Adolfo Piperno (1998): *Normalization of Typable Terms by Superdevelopments*. In: Georg Gottlob, Etienne Grandjean & Katrin Seyr, editors: *CSL, Lecture Notes in Computer Science* 1584. Springer, pp. 260–282.

[10] Jean-Jacques Lévy (1978): *Réductions correctes et optimales dans le lambda-calcul*. Ph.D. thesis, Paris VII.

[11] Jean-Jacques Lévy & Luc Maranget (1999): *Explicit Substitutions and Programming Languages*. In: C. Pandu Rangan, Venkatesh Raman & Ramaswamy Ramanujam, editors: *FSTTCS, Lecture Notes in Computer Science* 1738. Springer, pp. 181–200.

[12] Richard Mayr & Tobias Nipkow (1998): *Higher-Order Rewrite Systems and their Confluence*. *Theoretical Computer Science* 192, pp. 3–29.

[13] Oege de Moor & Ganesh Sittampalam (1998): *Generic Program Transformation*. In: *Advanced Functional Programming*. pp. 116–149.

[14] Oege de Moor & Ganesh Sittampalam (2001): *Higher-order matching for program transformation*. *Theor. Comput. Sci.* 269(1-2), pp. 135–162.

[15] Femke van Raamsdonk (1993): *Confluence and Superdevelopments*. In: Claude Kirchner, editor: *RTA, Lecture Notes in Computer Science* 690. Springer, pp. 168–182.

[16] Femke van Raamsdonk (1996): *Confluence and Normalisation for Higher-Order Rewriting*. Ph.D. thesis, Vrije Universiteit te Amsterdam.

[17] Ganesh Sittampalam & Oege de Moor (2001): *Higher-Order Pattern Matching for Automatically Applying Fusion Transformations*. In: Olivier Danvy & Andrzej Filinski, editors: *PADO, Lecture Notes in Computer Science* 2053. Springer, pp. 218–237.